

**REMARKS**

Claims 1-3, 5-14 and 16-22 are pending in the present application. By this Response, Claims 1, 7, 12, 18 and 20 are amended and claims 4 and 15 are canceled. Claims 1 and 20 are amended to incorporate subject matter similar to canceled claim 4. Claim 12 is amended to incorporate subject matter similar to canceled claim 15. Claims 7 and 18 are amended for clarity of subject matter. Reconsideration of the claims in view of the above amendments and the following remarks is respectfully requested.

**I. Non-Statutory Obviousness-Type Double Patenting**

The Office Action rejects claims 1-3, 5-14 and 16-22 under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claims 1-15 of U.S. Patent No. 6,433,794. Filed herein is a terminal disclaimer in relation to U.S. Patent No. 6,433,794. Therefore, the rejection of claims 1-3, 5-14 and 16-22 under the judicially created doctrine of obviousness-type double patenting has been overcome.

**II. 35 U.S.C. § 103, Alleged Obviousness, Claims 1-3, 5-14 and 16-22**

The Office Action rejects claims 1-3, 5-14 and 16-22 under 35 U.S.C. § 103(a) as being unpatentable over Hayes (U.S. Patent No. 6,205,476) in view of Walker et al. (U.S. Patent No. 6,332,218). This rejection is respectfully traversed.

As to claim 1, the Office Action states:

Hayes teaches client profile management by [col. 4, lines 8-33] providing through the browser [a web browser interface arranged to execute Java applications; col. 4, lines 15-21], an interface [configuration applet; col. 5, lines 45-67] in which allows for the configuration of the application by executing the application, configuring the application using the options provided by the application, and then saving the configuration [col. 4, lines 8-33], receiving a selection of classes through the interface [window...pops-up and allows the administrator to enter the path and file name of the text file containing the applet definitions to be appended; col. 18, lines 1-30], and storing the selection of classes [to save all pending changes, the administrator clicks on Files 1312 and the Save; col. 18, lines 1-30]. Hayes teaches a graphical interface that allows the user to

configure Java applications through a browser interface and but does not specifically teach using a selection of classes [list of applets] for use by a virtual machine.

However, Walker teaches allowing a user selecting classes [dependency list 26 includes one or more entries 30(1) through 30(P)...each of the entries 30(p), along with the contents of the respective fields 31 and 32(n), may be established in response to input information provided by an operator; col. 5, lines 21-53] and providing the selection of classes for use by a virtual machine [if the operator wishes to have a class 22 instantiated in the virtual machine 20 for use during processing of a program...he or she can enable the program/class loader 24 to instantiate the class 22 by providing and entry 26 for the program 21 in the dependency list 26 and identify the program in the field 31 and the class 22 in a field 32(n) in the entry 30(p); col. 5 lines 21-53].

Office Action dated October 9, 2003.

The present invention provides a method for selecting classes using a browser for use by a virtual machine in a data processing system. The browser provides an interface in which the interface allows for selection of classes for use by the virtual machine. A selection of classes is received through the interface. The selection of classes is stored by the browser and used by the browser when initializing the virtual machine.

Claim 1, which is representative of the other rejected independent claims 8, 12, 19 and 20 with regard to similarly recited subject matter, reads as follows:

1. A method for selecting classes using a browser for use by a virtual machine in a data processing system, the method comprising:
  - providing through the browser, an interface in which the interface allows for selection of classes for use by the virtual machine;
  - receiving a selection of classes through the interface;
  - storing the selection of classes, wherein the selection of classes is used by the browser when initializing the virtual machine; and
  - using the selection of the classes to initialize the virtual machine.(emphasis added)

Neither Hayes nor Walker, either alone or in combination, teach or suggest that, a selection of classes, which is received via an interface of a browser and stored, are used to initialize a virtual machine.

Hayes is directed to a system that provides the capability of allowing an administrator to configure a user application by running the application directly in the context of a user or user group, rather than in the context of the administrator. That is,

the configuration of the application is performed by executing the application, configuring the application using the options provided by the application for that purpose, and then saving the configuration as if the actual user or group were executing the application. However, there is no teaching or suggestion in Hayes regarding using a selection of classes to initialize a virtual machine.

The Office Action admits that Hayes does not teach or suggest this feature. However, the Office Action alleges that Walker teaches this feature at column 5, lines 21-53, which reads as follows:

The dependency list 26 includes one or more entries 30(1) through 30(P) (generally identified by reference numeral 30(p), each of which is associated with a respective one of the programs 21. Each entry 30(p) includes a program identifier field 31 and one or more class identifier fields 32(1) through 32(N) (generally identified by reference numeral 32(n)). The program identifier field 31 identifies one of the programs 21 which can be loaded in the virtual machine 20, and each of the class identifier fields 32(n) identifies one of the classes 22 which may be instantiated in the virtual machine. Each of the entries 30(p), along with the contents of the respective fields 31 and 32(n), may be established in response to input information provided by an operator in a conventional manner. Thus, the dependency list 26 allows the operator to indicate, when a program is loaded in the virtual machine 20, whether any classes 22 are to be instantiated in the virtual machine 20 along with the virtual machine, and if so, which classes. It will be appreciated that, if, when a program 21 is loaded in the virtual machine 20, no classes 22 are to be instantiated in the virtual machine 20 with the program 21, or if the program 21 itself enables all of the classes 22 that the operator wishes to have instantiated in the virtual machine 20 for use during processing of the program 21, the operator need not enable an entry 30(p) to be established for the program 21. However, if the operator wishes to have a class 22 instantiated in the virtual machine 20 for use during processing of a program, but the class 22 is not otherwise instantiated, he or she can enable the program/class loader 24 to instantiate the class 22 by providing an entry 26 for the program 21 in the dependency list 26 and identify the program in field 31 and the class 22 in a field 32(n) in the entry 30(p). (emphasis added)

This section of Walker merely teaches that a dependency list includes entries that are associated with a respective program and include a program identifier and a class identifier. The program identifier identifies the program, which can be loaded in the virtual machine, and the class identifier identifies the classes, which may be instantiated (or loaded) in the virtual machine. The dependency list allows the operator to indicate,

when a program is loaded in the virtual machine, whether any classes are to be instantiated in the virtual machine, and if so, which classes. However, there is nothing in this section, or any other section of Walker, that teaches using the selection of the classes to initialize the virtual machine. To the contrary, Walker teaches a mechanism for instantiating classes into an already initialized virtual machine. The classes are not used to initialize the virtual machine.

The Walker reference merely describes how classes maybe loaded with a program in a virtual machine that is already running. The classes are not used to initialize a virtual machine. While Walker teaches selection of classes, nowhere in the Walker reference is using the selection of classes to initialize a virtual machine even mentioned.

Moreover, there is not so much as a suggestion in either of the Hayes or Walker references to modify the references to include such a feature. That is, there is no teaching or suggestion in Hayes or Walker that a problem exists for which using the selection of classes to initialize a virtual machine is a solution. Walker only teaches loading programs and classes in a virtual machine that is already running and Hayes teaches configuring an application after the application is already running. Neither reference even recognizes the need or desirability to permit selection of classes for use in initializing a virtual machine and then to use the selection of classes to initialize the virtual machine.

One of ordinary skill in the art, being presented only with Hayes and Walker, and without having prior knowledge of Applicants' claimed invention, would not have found it obvious to combine and modify Hayes and Walker to arrive at Applicants' claimed invention. To the contrary, even if one were somehow motivated to combine Hayes and Walker, and it were somehow possible to combine the two systems, the result would not be the invention as recited in claim 1. The result would be an application that allows classes to be loaded in a virtual machine after it is running.

Thus, neither Hayes nor Walker teach or suggest the feature of using the selection of classes to initialize a virtual machine. There is not so much as a single teaching or suggestion in Hayes or Walker that would lead one of ordinary skill in the art to includes such a feature.

In view of the above, Applicants respectfully submit that neither Hayes or Walker, either alone or in combination, teach or suggest all of the features of claim 1, or the similar features found in independent claims 8, 12, 19 and 20. At least by virtue of their dependency on claims 1, 8, 12 and 20, neither Hayes or Walker, either alone or in combination, teach or suggest the features of dependent claims 2, 3, 5-7, 9-11, 13, 14, 16-18, 21 or 22. Accordingly, Applicants respectfully request withdrawal of the rejection of claims 1-3, 5-14 and 16-22 under 35 U.S.C. § 103(a).

Moreover, in addition to their dependency from independent claims 1, 8, 12 and 20, respectively, Hayes does not teach or suggest the specific features recited in claims 6, 7, 9, 17, 18 and 21. For example, with regard to claims 6, 17 and 21, Hayes does not teach or suggest where the selection of classes is a class path. The Office Action alleges that Hayes as modified teaches this feature at column 18, lines 19-29, which reads as follows:

The remaining fields are for optional information that may be required by the software upon invocation. A command button 1310, "Import Applet List from File", allows the administrator to append definitions of applets to the existing list 1306 from an existing text file. When button 1310 is clicked, the window shown in FIG. 14 pops-up and allows the administrator to enter the path and file name of the text file containing the applet definitions to be appended. To save all pending changes, the administrator clicks on File 1312 and then Save (not shown).

This section of Hayes merely teaches that an administrator may use a path and file name to identify a text file containing applet definitions. As shown above, the combination of Hayes and Walker does not teach using a selection of classes to initialize a virtual machine. Moreover, modifying Hayes by combining Hayes' teachings with Walker's teachings would not result in Hayes appending anything other than a text file containing applet definitions. Thus, Hayes as allegedly modified does not teach or suggest where the selection of classes is a class path, as the selection of classes is used to initialize a virtual machine.

As a further example, with regard to claims 7, 9 and 18, Hayes does not teach or suggest where the selection of classes causes a class path to be appended to a system class path. Once again, the Office Action alleges that Hayes as modified teaches this feature at column 18, lines 19-29, shown above. As shown above, the combination of

Hayes and Walker does not teach using a selection of classes to initialize a virtual machine and the modification of Hayes would not teach anything other than appending a text file containing applet definitions. Thus, Hayes as allegedly modified does not teach or suggest where the selection of classes causes a class path to be appended to a system class path.

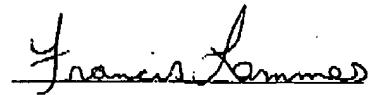
Therefore, in addition to being dependent on independent claims 1, 8, 12 and 20 respectively, dependent claims 6, 7, 9, 17, 18 and 21 are also distinguishable over Hayes by virtue of the specific features recited in these claims. Accordingly, Applicants respectfully request withdrawal of the rejection of claims 6, 7, 9, 17, 18 and 21 under 35 U.S.C. § 103(a).

### III. Conclusion

It is respectfully urged that the subject application is patentable over the Hayes and Walker and is now in condition for allowance. The Examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the Examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

Respectfully submitted,

DATE: January 7, 2004



Francis Lammes  
Reg. No. P-55,353  
Carstens, Yee & Cahoon, LLP  
P.O. Box 802334  
Dallas, TX 75380  
(972) 367-2001  
Agent for Applicants

SJW/fl